

JILID 1

BELAJAR BAHASA

C++

Dari seorang  
NEWBIE

AGUS WALUYO

## SEJARAH BAHASA C

Bahasa C merupakan bahasa pemrograman yang paling terkenal didunia dan mempunyai kemampuan lebih dari pada bahasa pemrograman yang lain. Bahasa C sendiri merupakan hasil buah karya dari Dennis Ritchi yang merupakan pengembangan dari bahasa BCPL yang telah ada lebih dahulu. Bahasa C merupakan *General purpose language* yaitu bahasa pemrograman yang dapat digunakan untuk tujuan apa saja. Diantaranya untuk pembuatan aplikasi pemrograman system, aplikasi cerdas (*artificial intelligence*), sistem pakar, utility, driver, database, browser, network programming, system operasi, game, virus dan lain-lain.

Diantara Kelebihan Bahasa C antara lain:

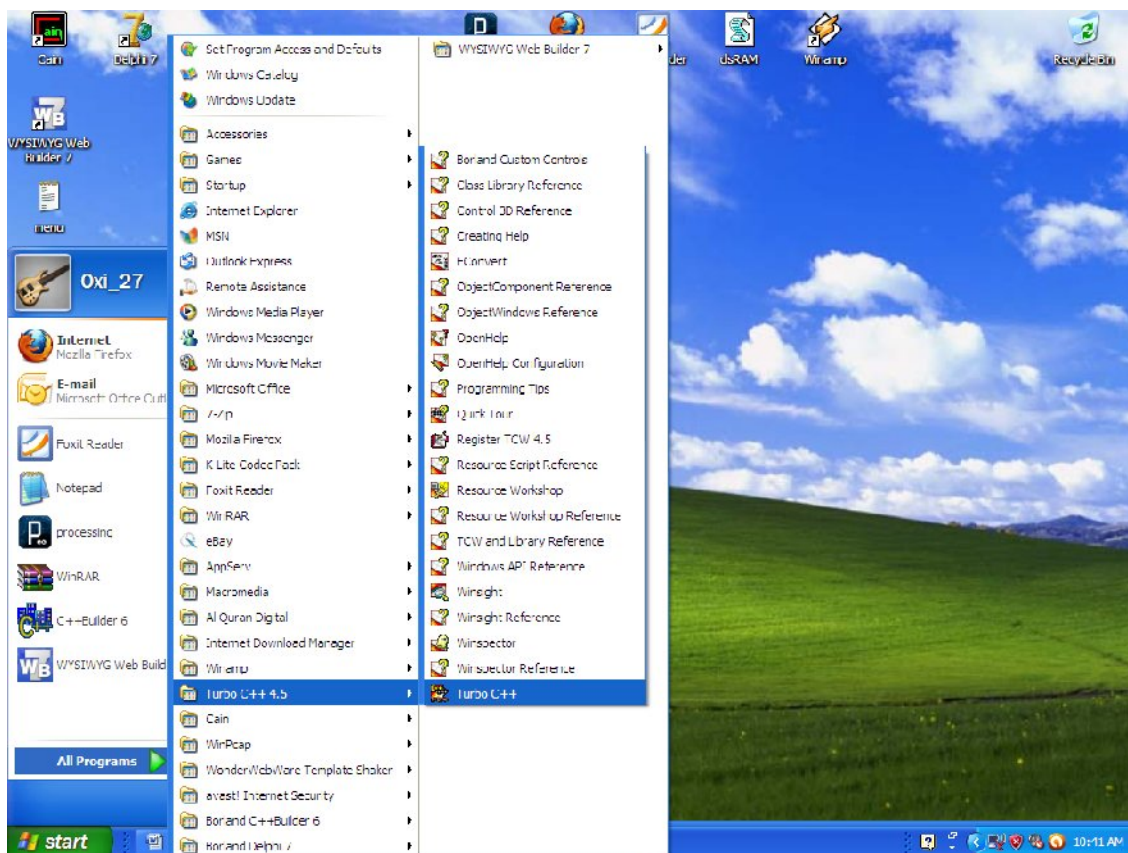
- ✓ **Pemrograman Berorientasi Onjek**
- ✓ **Portabilitas**
- ✓ **Singkat**

Tool yang akan kita gunakan pada kesempatan kali ini adalah **Turbo C++ 4.5**

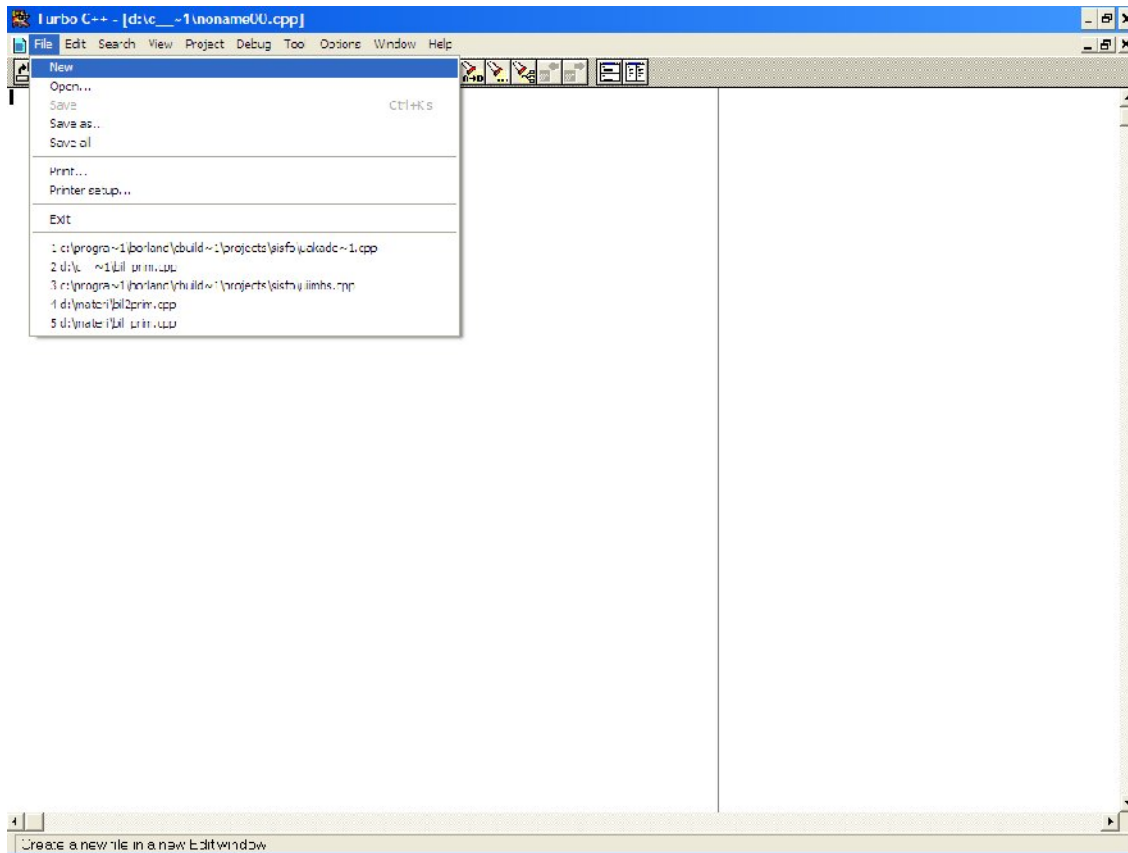
### Memulai menulis program

Langkah pertama untuk menulis program adalah membuka aplikasi Turbo c++ 4.5, langkah-langkahnya adalah sebagai berikut

1. Start
2. Program
3. Turbo c++ 4.5
4. turbo c++



langkah berikutnya adalah jika sudah masuk ke aplikasi Turbo C++ 4.5, Maka Klik file, pilih new, untuk membuat aplikasi baru



kemudian tulis source program pada form yang tersedia  
untuk meng-compile program gunakan kombinasi tombol **ALT+F9**, dan untuk menjalankan program gunakan kombinasi **CTRL + F9**  
untuk menyimpan program bisa menggunakan perintah **File** kemudian pilih **save as**.

## STRUKTUR PROGRAM BAHASA C++

Setiap bahasa pemrograman memiliki struktur yang unik yang membedakannya satu dengan yang lain, termasuk juga bahasa C++. Bentuk umum bahasa C++ adalah:

```
#include<file_header.h>
main()
{
//semua statement program
}
```

Contoh:

```
#include<iostream.h>// header file
#include<conio.h>// header file
void main()
{
cout<<"hello world";//statement untuk menampilkan
}
```

## KOMENTAR PROGRAM

C++ memberikan fasilitas kepada programmer untuk memberikan komentar pada source program. Fungsi komentar yaitu sebagai pengingat atau sebagai penjelas dari program. Komentar tidak akan decompile oleh mesin, sehingga tidak akan mempengaruhi program ketika diRUN.

Cara penulisan komentara ada dua macam, yaitu:

Dengan tanda `//` , ini khusus untuk komentar dalam satu baris

Dengan tanda `/*isi komentar*/`, untuk menuliskan komentar yang lebih dari satu baris

Contoh:

```
#include<iostream.h>// header file
#include<conio.h>// header file
void main()
{
cout<<"hello world";//ini contoh komentar
/*ini juga contoh komentar lho!!!!!!!!!!!!!!
Kzkzkzkzkzkzkzkzkzk*/
}
```

## IDENTIFIER DAN RESERVED WORD

**Identifier(pengenal)** adalah kata/nama yang diberikan oleh programmer kepada suatu variable, fungsi tipe data atau obyek-obyek lainnya yang didefinisikan oleh programmer

Aturan untuk mendefinisikan identifier adalah sebagai berikut:

1. Pengenal harus diawali dengan huruh(A,...,Z) atau karakter garis bawah(\_)
2. Selanjutnya boleh angka(0,...,9), karakter garis bawah(\_) atau tanda dolar(\$)
3. Panjang karakter boleh lebih dari 32 karakter, akan tetapi hanya 32 karakter pertama yang dianggap berarti

Contoh identifier

nama

Jumlah

Rata2

Harga\_jual\$

**Reserved Word** adalah sebuah kata yang telah digunakan sebagai kunci dalam bahasa C++ dan tidak bisa digunakan untuk sebuah identifier

contoh identifier dalam bahasa C++

Asm	Auto	Break	Case
default	char	const	continue
do	double	int	else
enum	extern	float	cdecl
far	for	goto	huge
if	interrupt	long	near
pascal	register	return	short
signed	sizeof	static	struct
switch	typedef	union	unsigned
void	volatile	while	namespace

dan masih banyak yang lainnya

## INPUT DAN OUTPUT

pada file preprosesor `#include<iostream.h>` maka fungsi I/O memakai

`cout<<` untuk menampilkan hasil kelayar

`cin>>` untuk memasukan data ke program

pada file preprosesor `#include<stdio.h>` maka fungsi I/O memakai

`printf(" ")` untuk menampilkan hasil kelayar

`scanf(" ")` untuk memasukan data ke program

Jadi didalam bahasa C++ banyak cara untuk melakukan proses I/O tergantung dari kebutuhan dan keinginan programer

contoh program 1

```
#include<iostream.h>

void main()
{
    int a;
    cout<<"masukan nilai a: ";
    cin>>a;
    cout<<"nilai a : "<<a;
}
```

contoh program 2

```
#include<stdio.h>

void main()
{
    int a;
    printf("masukan nilai a: ");
    scanf("%d",&a);
    printf("nilai a: %d",a);
}
```

## VARIABEL DAN KONSTANTA

**Variabel** adalah suatu peubah yang berfungsi menampung nilai yang dinamis, artinya nilai dari sebuah variabel dalam program dapat diubah-ubah ketika program tersebut dijalankan.

**Konstanta** adalah suatu tetapan yang bernilai konstan atau statis, artinya nilai dari suatu konstanta dalam suatu program tidak dapat diubah ketika program tersebut dijalankan.

cara mendeklarasikan variabel dan konstanta adalah sebagai berikut:

```
tipe_data nama_variabel;//deklarasi variabel
const tipe_data nama_konstanta;//deklarasi konstanta
```

contoh:

```
int a;
const int a=10;
```



## TIPE DATA

Tipe data digunakan untuk merepresentasikan nilai dari suatu variabel maupun konstanta. Tipe data sendiri dapat dikelompokkan menjadi dua, yaitu tipe data dasar dan tipe data bentukan

### Tipe data dasar

Tipe data	Besarnya memory	Jangkauan data
int	16 bit	-32768 s/d 32768
short int atau short	16 bit	-32768 s/d 32768
long int atau long	32 bit	-2147483648 s/d 2147483648
unsigned int	16 bit	0 s/d 65535
float	32 bit	3.4E-038 s/d 3.4E+038
double	64 bit	1.7E-038 s/d 1.7E+038
long double	64 bit	1.7E-038 s/d 1.7E+038
char	8 bit	-128 s/d 127
unsigned char	8 bit	0 s/d 255
enum	16 bit	-32768 s/d 32768
unsigned long	32 bit	0 s/d 4294967295
long	32 bit	-2147483648 s/d 2147483647

### Tipe data Bentukan

**Tipe data bentukan** adalah tipe-tipe yang dibuat sendiri oleh oleh programmer dengan menggunakan tipe-tipe dasar yang tersedia.

## OPERATOR

Operator merupakan simbol atau kata yang dapat digunakan dalam program untuk operasi atau manipulasi pada data, operator dalam bahasa C++ dapat dikelompokkan sebagai berikut

1. Operator pengerjaan (*assignment operators*)
2. Operator aritmetika (*arithmetic operators*)
3. Operator peningkatan dan penurunan (*increment and decrement operators*)
4. Operator hubungan (*relational operators*)
5. Operator logika (*logical operators*)

selanjutnya mari kita bahas satu persatu

1. Operator pengerjaan (*assignment operators*)

**Operator pengerjaan (*assignment operator*)** berupa tanda sama dengan (=) yang digunakan untuk mengisi nilai yang berada disebelah kanan tanda ke dalam kiri tanda

contoh

a=1

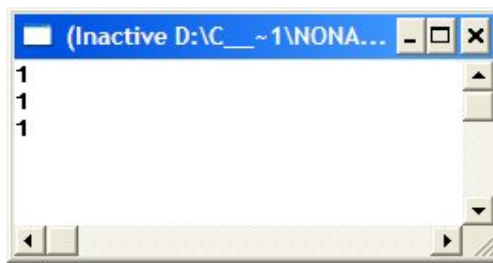
c=b=a

menunjukkan nilai a diisikan ke b kemudian nilai b diisikan ke c

contoh dalam program

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int a=1,b,c;
    c=b=a;
    cout<<a<<endl;//endl berfungsi untuk membuat baris baru/meng-enter
    cout<<b<<endl;
    cout<<c;
}
```

hasil runing program sebagai berikut



## 2. Operator aritmetika (arithmetic operators)

**Operator aritmetika (arithmetic operators)** adalah operator yang digunakan untuk memanipulasi kalkulasi matematika

tabel operator

Operator	Keterangan
*	Perkalian
/	Pembagian
%	Modulus atau sisa bagi
+	Pertambahan
-	Pengurangan

contoh program 1

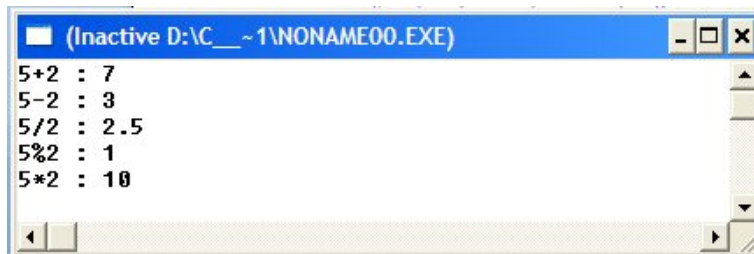
```
#include<iostream.h>
#include<conio.h>
void main()
{
    float tambah,kurang,bagi,kali,a=5,b=2;
    int sisa;
```



```
tambah=a+b;
kurang=a-b;
bagi=a/b;
kali=a*b;
sisasisa=5%2;

cout<<"5+2 : "<<tambah<<endl;
cout<<"5-2 : "<<kurang<<endl;
cout<<"5/2 : "<<bagi<<endl;
cout<<"5%2 : "<<sisasisa<<endl;
cout<<"5*2 : "<<kali<<endl;
}
```

hasil run program adalah sebagai berikut



```
(Inactive D:\C__~1\NONAME00.EXE)
5+2 : 7
5-2 : 3
5/2 : 2.5
5%2 : 1
5*2 : 10
```

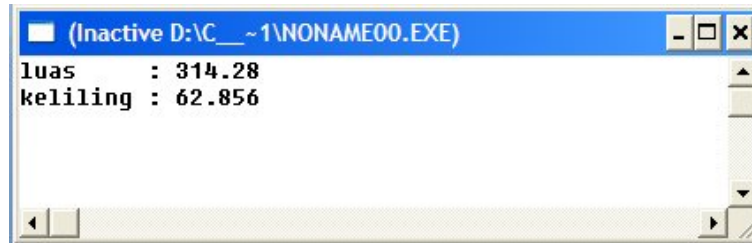
contoh program 2

menghitung luas dan keliling lingkaran

```
#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
float luas,pi,jari,keliling;
pi=3.1428;
jari=10;

luas=pi*jari*jari;
keliling=2*pi*jari;
cout<<"luas : "<<luas<<endl;
cout<<"keliling : "<<keliling<<endl;
}
```

hasil run program adalah sebagai berikut



```
(Inactive D:\C__~1\NONAME00.EXE)
luas : 314.28
keliling : 62.856
```

3. Operator peningkatan dan penurunan (increment and decrement operators)

**Operator peningkatan (++)** adalah operator yang digunakan untuk menambah nilai 1 pada nilai sebelumnya, sedangkan **operator penurunan (--)** adalah operator yang digunakan untuk mengurangi nilai 1 pada nilai sebelumnya

contoh program

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int awal,incre,decre;
    awal=10;
    incre=++awal;
    decre=--awal;

    cout<<"nilai awal : "<<awal<<endl;
    cout<<"setelah increment: "<<incre<<endl;
    cout<<"setelah decrement: "<<decre;
}
```

hasil runing program adalah sebagai berikut



```
(Inactive D:\C__~1\NONAME00.EXE)
nilai awal : 10
setelah increment: 11
setelah decrement: 10
```

4. Operator hubungan (relational operators)

**Operator hubungan (relational operators)** adalah operator yang digunakan untuk membandingkan dua buah elemen nilai dan akan menghasilkan dua kemungkinan yaitu benar(1) atau salah (0)

tabel relational operator

Operator	Keterangan
>	Lebih dari
<	Kurang dari
>=	Lebih dari sama dengan
<=	Kurang dari sama dengan
==	sama dengan
!=	tidak sama dengan

contoh dalam program

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int a=2,b=3,c=3;
    cout<<"a : "<<a<<endl;
    cout<<"b : "<<b<<endl;
    cout<<"c : "<<c<<endl;
    cout<<"a<b:"<<(a<b)<<endl;
    cout<<"a>b:"<<(a>b)<<endl;
    cout<<"a==c:"<<(a==c)<<endl;
    cout<<"b==c:"<<(b==c)<<endl;
    cout<<"a!=c:"<<(a!=c);
}
```

hasil runing program sebagai berikut

```
(Inactive D:\C__~1\NONAME00.EXE)
a : 2
b : 3
c : 3
a<b:1
a>b:0
a==c:0
b==c:1
a!=c:1
```

##### 5. Operator logika (logical operators)

**Operator logika (logical operators)** digunakan dalam operasi-operasi logika, yaitu operasi yang menghasilkan nilai logic, *true* atau *false*(0 atau 1 ). Operator logika selalu berdampingan dengan operator hubungan. karena biasanya operator logika membandingkan nilai (true atau false) dari beberapa operator hubungan

tabel operator logika

Operator	Keterangan
&&	Operator logika AND
	Operator logika OR
!	Operator logika NOT

contoh program

```
#include<iostream.h>
#include<conio.h>
void main()
{
    int tugas,uts,uas;
    tugas=78;
    uts=90;
    uas=89;
    if(tugas>70&&uts>75&&uas>75){
        cout<<"lulus";
    }
    else{
        cout<<"tidak lulus";
    }
}
```

hasil run program adalah sebagai berikut  
lulus

### soal dan penyelesaian

1. buatlah program untuk menghitung luas dan volume suatu balok, rumus  $luas = (2 * p * l) + (2 * p * t) + (2 * l * t)$ , sedangkan volume balok  $= p * l * t$ . dengan  $p$ =panjang,  $l$ =lebar,  $t$ =tinggi.  
jawab

```
#include<iostream.h>
void main(){
    float panjang,lebar,tinggi,volume,luas;
    cout<<"=====Menu Balok===== "<<endl;
        cout<<"panjang : ";cin>>panjang;
        cout<<"lebar : ";cin>>lebar;
        cout<<"tinggi : ";cin>>tinggi;
        volume=panjang*lebar*tinggi;
        luas=2*((panjang*lebar)+(panjang*tinggi)+(lebar*tinggi));
        cout<<"volume : "<<volume<<endl;
        cout<<"luas : "<<luas<<endl;
```

```
}
```

2. buatlah program untuk mengkonversi nilai mata uang dengan input rupiah, dengan rumus  
dolar=rupiah/8150  
ringgit=rupiah/2500  
yen=rupiah/4000  
jawab

```
#include<iostream.h>
void main(){
float rupiah,dolar,ringgit,yen;
cout<<"====Konversi Mata uang===="<<endl;
cout<<"masukan mata uang rupiah: ";cin>>rupiah;
dolar=rupiah/8150;
ringgit=rupiah/2500;
yen=rupiah/4000;
cout<<"====="<<endl;
cout<<"Dolar : "<<dolar<<endl;
cout<<"Ringgit: "<<ringgit<<endl;
cout<<"Yen   : "<<yen<<endl;
cout<<"====Terima Kasih====";
}
```

## DECISION atau PERCABANGAN

**Decision atau percabangan** adalah suatu perintah untuk memilih yang didasarkan atas kondisi tertentu. Suatu aksi akan dipilih hanya jika kondisi yang telah didefinisikan terpenuhi. dalam bahasa C++ ada dua perintah percabangan yaitu perintah if dan perintah switch.

### Perintah If

perintah if adalah perintah yang digunakan untuk memilih satu dari dua atau lebih alternatif jawaban yang tersedia. if terdiri beberapa varian, diantaranya if, if...else dan nested if

1. if  
bentuk umumnya adalah  
if (kondisi){  
pernyataan-pernyataan;  
}

contoh program

```
#include<iostream.h>
#include<conio.h>
void main(){
int a;
cout<<"masukan nilai a : ";
cin>>a;
if(a>50){
cout<<"a lebih dari 50";
}
}
```

hasil running program adalah sebagai berikut



2. if else  
perintah if..else mempunyai bentuk umum sebagai berikut:

```
if (kondisi)
pernyataan;

else
pernyataan;
```

Arti dari bentuk ini adalah jika kondisi benar maka pernyataan yang berada tepat setelah if yang akan dikerjakan, jika kondisi salah maka pernyataan yang berada setelah else yang akan dikerjakan

contoh program 1

```
#include<iostream.h>
#include<conio.h>
void main(){
int a;
cout<<"masukan nilai a : ";
cin>>a;
if(a>50)
cout<<a<<" lebih dari 50";

else
cout<<a<<" kurang dari 50";
}
```

hasil setelah dijalankan sebagai berikut

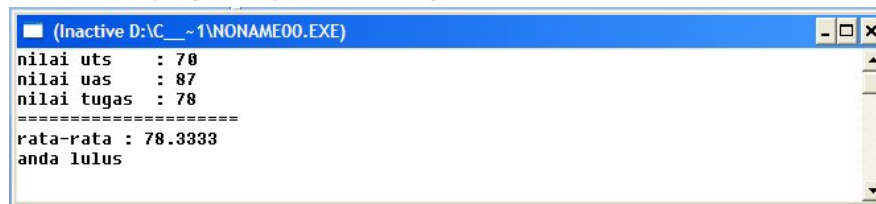


contoh program 2

```
#include<iostream.h>
#include<conio.h>
void main(){
clrscr();
float uts,uas,tugas,rata2;
cout<<"nilai uts  : ";cin>>uts;
cout<<"nilai uas  : ";cin>>uas;
cout<<"nilai tugas : ";cin>>tugas;
rata2=(uts+uas+tugas)/3;
cout<<"====="<<endl;
cout<<"rata-rata : "<<rata2<<endl;
if(rata2>70)
cout<<"anda lulus";
else
cout<<"anda tidak lulus";
}
```



hasil setelah program dijalankan sebagai berikut



```
(Inactive D:\C__~1\NONAME00.EXE)
nilai uts   : 70
nilai uas   : 87
nilai tugas : 78
=====
rata-rata   : 78.3333
anda lulus
```

### 3. Nested if

Struktur percabangan jenis ini memiliki tiga buah kondisi atau lebih. bentuk umum dari struktur ini adalah sebagai berikut

```
if(kondisi1)
    pernyataan;
else if(kondisi2)
    pernyataan;
else if(kondisi3)
    pernyataan;
....
else if(kondisi ke-n)
    pernyataan;
else
    pernyataan;
```

contoh program

```
#include<iostream.h>
#include<conio.h>
void main(){
    clrscr();
    float uts,uas,tugas,rata2;
    char grade;
    cout<<"===== "<<endl;
    cout<<"=== |Masukan Nilai| === "<<endl;
    cout<<"===== "<<endl;
    cout<<"nilai uts   : ";cin>>uts;
    cout<<"nilai uas   : ";cin>>uas;
    cout<<"nilai tugas : ";cin>>tugas;
    rata2=(uts+uas+tugas)/3;
    cout<<"===== "<<endl;

    if(rata2>85){
        grade='A';
    }

    else if(rata2>75){
```

```

        grade='B';
    }

    else if(rata2>65){
        grade='C';
    }

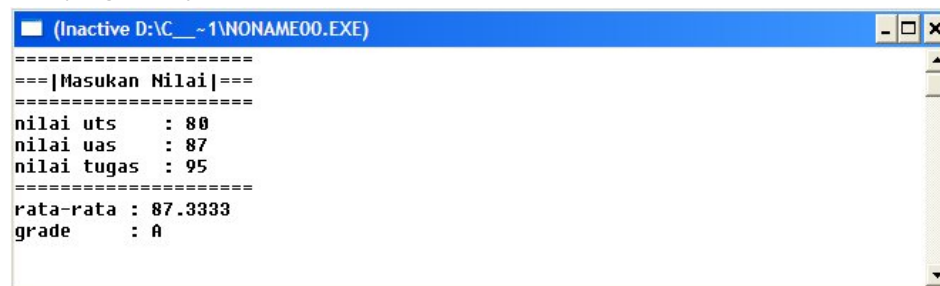
    else if(rata2>50){
        grade='D';
    }

    else{
        grade='E';
    }

    cout<<"rata-rata : "<<rata2<<endl;
    cout<<"grade   : "<<grade;
}

```

hasil program dijalankan



```

(Inactive D:\C__ ~ 1\NONAME00.EXE)
=====
---|Masukan Nilai|---
=====
nilai uts   : 80
nilai uas   : 87
nilai tugas : 95
=====
rata-rata : 87.3333
grade      : A

```

## SWITCH

**perintah switch** adalah perintah pemilihan yang melibatkan banyak kondisi, biasanya dipakai untuk menggantikan perintah if yang terlalu banyak. bentuk umum dari switch adalah sebagai berikut

```

switch(kondisi){
case konstanta_1: statment; break;
case konstanta_2: statment; break;
case konstanta_3: statment; break;
case konstanta_n: statment; break;
default: statment_lainnya; break;
}

```

contoh program 1

```

#include<iostream.h>
#include<conio.h>
void main()

```

```

{
clrscr();
int pilihan;
cout<<"masukan pilihan :";
cin>>pilihan;
switch(pilihan){
case 1: cout<<"pilihan 1";break;
case 2: cout<<"pilihan 2";break;
case 3: cout<<"pilihan 3";break;
default: cout<<"pilihan tidak ada";break;
}
}

```

contoh program 2

```

#include<iostream.h>
#include<conio.h>
void main()
{
clrscr();
float sisi,panjang,lebar,tinggi,jari,volume,luas,keliling;
int pil;
cout<<"=====PILIHAN PERHITUNGAN===== "<<endl;
cout<<"1. volume dan luas kubus"<<endl;
cout<<"2. volume dan luas balok"<<endl;
cout<<"3. Keliling dan luas lingkaran"<<endl;
cout<<"===== "<<endl;
cout<<" masukan pilihan [1,2,3] : ";cin>>pil;
switch(pil){
case 1:cout<<"=====Menu Kubus===== "<<endl;
        cout<<"panjang sisi : ";cin>>sisi;
        volume=sisi*sisi*sisi;
        luas=6*(sisi*sisi);
        cout<<"volume      : "<<volume<<endl;
        cout<<"luas        : "<<luas;
        break;
case 2:cout<<"=====Menu Balok===== "<<endl;
        cout<<"panjang : ";cin>>panjang;
        cout<<"lebar  : ";cin>>lebar;
        cout<<"tinggi : ";cin>>tinggi;
        volume=panjang*lebar*tinggi;
        luas=2*((panjang*lebar)+(panjang*tinggi)+(lebar*tinggi));
        cout<<"volume : "<<volume<<endl;
        cout<<"luas   : "<<luas<<endl;

```

```

        break;
    case 3: cout<<"====Menu Lingkaran===="<<endl;
        cout<<"jari-jari : ";cin>>jari;
        luas=3.14*jari*jari;
        keliling=2*3.14*jari;
        cout<<"luas    : "<<luas<<endl;
        cout<<"keliling : "<<keliling;
        break;
    default: cout<<"pilihan salah";
        break;
    }
}

```

hasil program setelah dijalankan adalah sebagai berikut

```

(Inactive D:\C__~1\NONAME00.EXE)
=====PILIHAN PERHITUNGAN=====
1. volume dan luas kubus
2. volume dan luas balok
3. Keliling dan luas lingkaran
=====
    masukan pilihan [1,2,3] : 2
=====Menu Balok=====
panjang : 10
lebar   : 12
tinggi  : 5
volume  : 600
luas    : 460

```

## PERULANGAN

**Perulangan atau looping** adalah perintah yang digunakan untuk melakukan proses secara berulang-ulang jika suatu kondisi dipenuhi atau sebaliknya. dalam bahasa C++ ada tiga macam perulangan, yaitu FOR, WHILE dan DO..WHILE

### 1. For

perintah FOR digunakan untuk mengulang statment yang sudah pasti berapa kali pengulangannya dengan kata lain berapa kali statment akan diulangi sudah diketahui dan didefinisikan kedalam perintah FOR. bentuk umum dari perintah FOR adalah sebagai berikut

```

for (variabel_awal=nilai_awal;kondisi;iterasi)
{
    pernyataan/statment;
}

```

contoh program

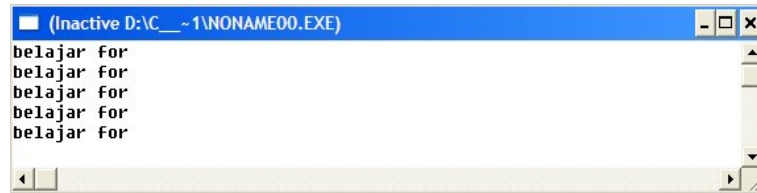
```

#include<iostream.h>
#include<conio.h>

```

```
void main()
{
    clrscr();
    int a;
    for(a=0;a<5;a++){
        cout<<"belajar for"<<endl;
    }
}
```

apabila program dijalankan akan menghasilkan



```
(Inactive D:\C__-1\NONAME00.EXE)
belajar for
belajar for
belajar for
belajar for
belajar for
```

contoh program 2

```
#include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    int a;
    for(a=1;a<=5;a++){
        cout<<a<<endl;
    }
}
```

bila program dijalankan, maka hasilnya sebagai berikut



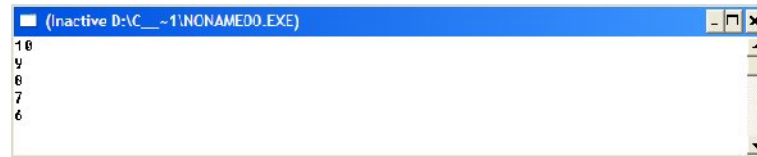
```
(Inactive D:\C__-1\NONAME00.EXE)
1
2
3
4
5
```

contoh program 3

```
#include<iostream.h>
#include<conio.h>
void main()
{
    clrscr();
    int a;
    for(a=10;a>5;a--){
```

```
cout<<a<<endl;
}
}
```

bila program dijalankan, maka hasilnya sebagai berikut



```
(Inactive D:\C__ - 1\NONAME00.EXE)
10
9
8
7
6
```

contoh program 4

menampilkan bilangan bulat dari 1-10

```
#include<iostream.h>
void main(){
int a;
for(a=1;a<=10;a++)
{
    if(a%2==0)
    {
        cout<<a<<endl;
    }
}
}
```

jika program dijalankan maka hasilnya sebagai berikut



```
(Inactive D:\C__ - 1\NONAME00.EXE)
2
4
6
8
10
```

contoh program 5

membuat segitiga piramid dengan nilai tinggi yang berubah-ubah

```
#include<iostream.h>
void main(){
int lebar,tinggi,n;
cout<<"masukan tinggi segitiga: ";cin>>n;
for(lebar=1;lebar<=n;lebar++){
for(tinggi=1;tinggi<=lebar;tinggi++)
cout<<"*";
cout<<endl;
}
}
```

hasil penjalanan program adalah sebagai berikut



```
(Inactive D:\C_1\NONAME00.EXE)
masukan linqqi sequitiqa: 5
*
**
***
****
*****
```

## 2. While

Struktur jenis ini melakukan pengecekan kondisi pada bagian awal, artinya jika kondisi terpenuhi maka perulangan baru akan dijalankan namun jika kondisi tidak terpenuhi maka perulangan tidak akan pernah dijalankan. bentuk umum dari perintah ini adalah sebagai berikut

```
nama_variabel_while=nilai_awal;
while(kondisi)
{
    statment_yang_akan_diulangi;
    nama_variabel_while++/--;
}
```

contoh program 1

```
#include<iostream.h>
void main(){
    int a=0;
    while(a<5){
        cout<<"belajar while"<<endl;
        a++;
    }
}
```

bila program dijalankan, maka hasilnya sebagai berikut



```
(Inactive D:\C_1\NONAME00.EXE)
belajar while
belajar while
belajar while
belajar while
belajar while
```

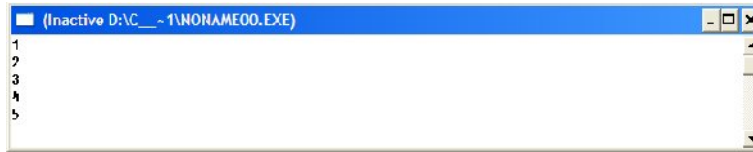
contoh program 2

```
#include<iostream.h>
void main(){
    int a=1;
    while(a<=5){
        cout<<a<<endl;
        a++;
    }
```



```
}
}
```

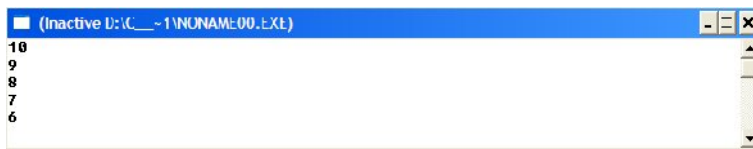
bila program dijalankan, maka hasilnya sebagai berikut



contoh program 3

```
#include<iostream.h>
void main(){
int a=10;
while(a>5){
cout<<a<<endl;
a--;
}
}
```

bila program dijalankan, maka hasilnya sebagai berikut



### 3. Do..While

Perintah do..while pada dasarnya sama dengan perintah while, namun untuk perintah do..while, pengecekan kondisi dilakukan setelah statment yang akan diulangi. bentuk umum dari do..while adalah sebagai berikut

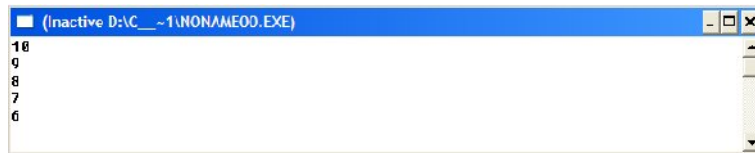
```
nama_variabel=nilai_awal;
do{
statment_yang_akan_diulangi;
nama_variabel--/++;
}while(kondisi)
```

contoh program 1

```
#include<iostream.h>
void main(){
int a=10;
do
```

```
{  
cout<<a<<endl;  
a--;  
}while(a>5);  
}
```

jika program dijalankan, maka hasilnya sebagai berikut



```
(Inactive D:\C__-1\NONAME00.EXE)  
10  
9  
8  
7  
6
```

contoh program 2

```
#include<iostream.h>  
void main(){  
int a=1;  
char ulang;  
do  
{  
cout<<a<<endl;  
a++;  
cout<<"ulangi looping ?: ";cin>>ulang;  
}while(ulang=='y');  
}
```

jika program dijalankan maka hasilnya sebagai berikut



```
(Inactive D:\C__-1\NONAME00.EXE)  
1  
ulangi looping ?: y  
2  
ulangi looping ?: y  
3  
ulangi looping ?: y  
4  
ulangi looping ?: y  
5  
ulangi looping ?: t
```

## LARIK atau ARRAY

**Array** atau yang biasa disebut sebagai larik adalah kumpulan atau koleksi data yang disimpan dengan nama yang sama dan memiliki tipe data yang sama dan setiap elemen diakses dengan membedakan indeks arraynya. jadi larik adalah kumpulan data dengan tipe yang sama yang disimpan kedalam satu variabel namun berbeda indeksnya,

nilai[0]	70
nilai[1]	60
nilai[2]	79
nilai[3]	80
nilai[4]	75

dari tabel diatas dapat dijelaskan bahwa array tersebut bernama nilai yang memiliki 5 lokasi memori yang semua tipenya int. masing-masing dari nilai memiliki identitas yang serupa yaitu nilai, namun indeksnya berbeda-beda dimulai 0(nol). indeks dituliskan didalam kurung siku '[...]'. sebagai contoh 79, adalah isi dari variabel **nilai** pada indeks ke [2].

### DEKLARASI ARRAY

array dideklarasikan dengan mencantumkan tipe data, nama variabel, serta berapa banyak lokasi memori yang dipesan atau yang ingin dibuat misalkan

```
float rata2[10];
```

artinya kita hendak memesan array bertipe float dengan nama rata2 sebanyak 10 lokasi memori pada pendeklarasian array kita juga bisa langsung mengisi/menginisialisasi nilai array pada masing-masing indeks yang kita inginkan.

contoh

```
int nilai[5]={75,70,90,80,85}
```

artinya kita memesan array dengan nama nilai dan bertipe integer dengan banyak lokasi 5. dengan masing-masing sudah terinisialisasi pada

nilai[0]	75
nilai[1]	70
nilai[2]	90
nilai[3]	80
nilai[4]	85

### PEMANGGILAN ARRAY

array dapat dipanggil dengan cara menuliskan nama variabel disertai indeks keberapa yang ingin dipanggil, misalkan untuk tabel dibawah ini

nilai[0]	75
nilai[1]	70

nilai[2]	90
nilai[3]	80
nilai[4]	85

jika kita menuliskan diprogram

```
cout<<"nilai[1]";
```

maka program akan memanggil isi dari nilai yang berada pada indeks dengan nomor 1, yaitu 70.

contoh program 1

```
#include<iostream.h>
void main()
{
    int a,nilai[6];
    for(a=1;a<=5;a++){
        cout<<"masukan nilai kedalam array ke-"<<a<<": ";
        cin>>nilai[a];
    }
    cout<<"===== "<<endl;
    cout<<"lihat array secara descending"<<endl;
    for(int b=5;b>0;b--){
        cout<<"isi dari array ke-"<<b<<": "<<nilai[b]<<endl;
    }
}
```

apabila program dijalankan maka hasilnya sebagai berikut

```
(Inactive D:\C_~1\NONAME00.EXE)
masukan nilai kedalam array ke-1: 12
masukan nilai kedalam array ke-2: 32
masukan nilai kedalam array ke-3: 45
masukan nilai kedalam array ke-4: 32
masukan nilai kedalam array ke-5: 456
=====
lihat array secara descending
isi dari array ke-5 : 456
isi dari array ke-4 : 32
isi dari array ke-3 : 45
isi dari array ke-2 : 32
isi dari array ke-1 : 12
```

## ARRAY DUA DIMENSI

adalah array yang terdiri dari dua elemen indeks. Sebagai contoh matrik A berukuran 3x2 dapat direpresentasikan kedalam array dua dimensi dengan bentuk  $A[3][2]=\{\{5,3\},\{10,12\},\{4,5\}\}$ , selengkapnya bisa dilihat di dalam tabel

	0	1
0	5	3
1	10	12
2	4	5

definisi variabel untuk setiap elemen tersebut adalah

	0	1
0	A[0][0]	A[0][1]
1	A[1][0]	A[1][1]
2	A[2][0]	A[2][1]

atau bila ditransformasikan kedalam program hasilnya sebagai berikut

```
#include<iostream.h>
#include<conio.h>
void main(){
clrscr();
int A[3][2]={5,3},{10,12},{4,5}};
    for(int b=0;b<3;b++){
        for(int c=0;c<2;c++){
            cout<<"A["<<b<<"]["<<c<<"] : "<<A[b][c]<<endl;
        }
    }
}
```

hasil program setelah dijalankan adalah sebagai berikut



```
(Inactive D:\C__~1\NONAME00.EXE)
A[0][0] : 5
A[0][1] : 3
A[1][0] : 10
A[1][1] : 12
A[2][0] : 4
A[2][1] : 5
```

## GOTOXY

**GOTOXY** adalah perintah untuk meletakkan perintah output program pada suatu lokasi (x,y)

contoh program

```
#include<iostream.h>
#include<conio.h>
void main(){
clrscr();
gotoxy(10,10);cout<<"ini letak gotoxy(10,10)";
}
```



```
(Inactive D:\C__~1\NONAME00.EXE)

ini letak gotoxy(10,10)
```

## Soal dan Penyelesaian

1. Buatlah program untuk menghitung berat ideal seseorang dengan rumus  
Berat ideal=tinggi-110.

jika berat kurang dari berat ideal-3, maka dinyatakan kurus

jika berat lebih dari berat ideal+3, maka dinyatakan gemuk.

jawab

```
#include<iostream.h>
void main()
{
    int t,b,ideal,a;
    cout<<"===== "<<endl;
    cout<<"    | "<<endl;
    cout<<"===== "<<endl;
    cout<<"tinggi : ";
    cin>>t;
    cout<<"berat : ";
    cin>>b;
    cout<<"===== "<<endl<<endl;;
    ideal=t-110;
    a=b-ideal;
    if(a>=3)
    {
        cout<<"berat anda tidak ideal"<<endl;
        cout<<"anda kelebihan : "<<(a-3)<<" kilogram";
    }
    else
    {
        if(a<=-3)
        {
            cout<<"berat anda tidak ideal"<<endl;
            cout<<"anda kekurangan : "<<(a-3)<<" kilogram";
        }
        else
        {
            cout<<"berat anda ideal";
        }
    }
}
```

2. buatlah program untuk menghitung total gaji dari 5 buah karyawan dengan input berupa NIP, nama, gaji. Kemudian cari total gajinya. Gunakanlah perintah gotoxy untuk memperindah tampilan

jawab

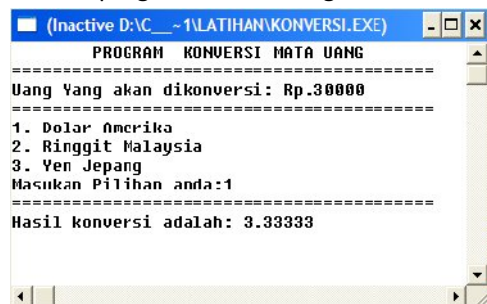
```

#include<iostream.h>
#include<conio.h>
char nama[10][5];
char NIP[10][5];
long gaji[5];

void main()
{
clrscr();
long total_gaji;
total_gaji=0;
cout<<"===== "<<endl;
cout<<"NO NIP      nama      gaji"<<endl;
cout<<"===== "<<endl;
for(int a=1;a<=5;a++)
{
cout<<a<<endl;
gotoxy(5,3+a);cin>>NIP[a];
gotoxy(22,3+a);cin>>nama[a];
gotoxy(38,3+a);cin>>gaji[a];
}
cout<<"===== "<<endl;
for(a=1;a<=5;a++){
total_gaji=total_gaji+gaji[a];
}
cout<<"          Total gaji : "<<total_gaji;
}

```

3. buatlah program untuk mengkonversi mata uang dengan tampilan sebagai berikut



Dolar=rupiah/9000  
Ringgit=rupiah/2500  
yen=rupiah/500  
jawab

```

#include<iostream.h>
#include<conio.h>
main()

```



```
{
clrscr();
int pilihan;
float hasil,uangawal;
cout<<"    PROGRAM KONVERSI MATA UANG"<<endl;
cout<<"===== "<<endl;
cout<<"Uang Yang akan dikonversi: Rp.";cin>>uangawal;
cout<<"===== "<<endl;
cout<<"1. Dolar Amerika"<<endl;
cout<<"2. Ringgit Malaysia"<<endl;
cout<<"3. Yen Jepang"<<endl;
cout<<"Masukan Pilihan anda:";cin>>pilihan;
cout<<"===== "<<endl;
if(pilihan==1)
{
hasil = uangawal/9000;
cout<<"Hasil konversi adalah: "<<hasil;
}
else if(pilihan==2)
{
hasil = uangawal/2500;
cout<<"Hasil konversi adalah: "<<hasil;
}
else
{
hasil = uangawal/500;
cout<<"Hasil konversi adalah: "<<hasil;
}
getch ();
return 0;
}
```

**EVALUASI**

1. Buatlah Program untuk menampilkan keluaran sebagai berikut, dengan nilai n yang dapat diubah –ubah?

```
****
***
**
*

untuk n=4;
*****
*****
****
***
**
*

untuk n=6;
```

2. Buatlah program untuk menampilkan deret bilangan prima 1 sampai dengan n, dengan nilai n dimasukkan oleh user?,  
misalkan n=7, maka outputnya 2,3,5,7  
n=20, maka outputnya 2,3,5,7,11,13,17,19

3. Buatlah program untuk mencari nilai tertinggi dan terendah dari 6(enam) data yang dimasukan oleh user, dengan syarat bilangan terkecil harus bilangan ganjil, dan bilangan terbesar harus bilangan genap  
misalka kita masukan 10,87,13,46,17,15, maka akan diperoleh  
nilai tertinggi=46  
nilai terendah=13

4. Buatlah program untuk mencari nilai tertinggi atau terendah dari 7 data masukan, dengan ketentuan jika ada bilangan prima dalam masukan,maka nilai maksimal harus bilangan prima. jika tidak ada bilangan prima dalam masukan, maka cari nilai minimal dari bilangan yang ada.  
contoh 1 data masukan 10,11,19,4,17,20,27, maka outputnya, nilai maksimal=19  
contoh 2 data masukan 15,10,9,27,12,6,33, maka outputnya, nilai minimal=6

5. Buatlah program dengan tampilan output sebagai berikut  
Dengan ketentuan  
masukan jenis obat: menunjukan berapa jenis obat yang dibeli dan dimasukan sebagai input  
kode barang, nama obat dan harga: boleh diubah sesuai keinginan programmer  
kode barang: dimasukan sebagai input. sedangkan nama obat dan harga menyesuaikan dengan kode barang yang dimasukkan  
jumlah: dimasukan sebagai input  
total = harga\*jumlah

total bayar=  $\Sigma$ total

dibayar=dimasukan sebagai input

kembali=dibayar-total bayar

Nota Penjualan					
Apotek FARMA MEDIKA					
=====					
masukan jenis obat : 4					
=====					
no	kode barang	nama obat	harga	jumlah	total
1	2	antangan	1700	12	20400
2	4	extrajoss	2000	1	2000
3	3	kukubima	1500	5	7500
4	5	tolak angin	500	4	2000
=====					
				total bayar	: 31900
				dibayar	: 50000
				kembali	: 18100
=====					
				TERIMA KASIH	
hitung lagi [y/t]:					

6. Buatlah program dengan tampilan input sebagai berikut

```

Tampilan input ke- 1
Nilai Test Siswa Kursus
=====
No Urut siswa      : 1
Nama Siswa         : eko
Jumlah test yang diikuti : 3
=====
Nilai Test 1      : 80
Nilai Test 2      : 70
Nilai Test 3      : 67
input data siswa lagi<y/t> : y_

Tampilan input ke- 2
Nilai Test Siswa Kursus
=====
No Urut siswa      : 2
Nama Siswa         : Fitri
Jumlah test yang diikuti : 2
=====
Nilai Test 1      : 67
Nilai Test 2      : 80
input data siswa lagi<y/t> : t_

```

dan outputnya sebagai berikut

```

Laporan Hasil Nilai
Rata-rata Nilai Test Siswa kursus
=====
no urut      Nama Siswa      rata-rata nilai
=====
1            eko            72.3333
2            Fitri          78
=====

```

dengan ketentuan

nama siswa, jumlah tes yang diikuti, nilai test serta input data siswa lagi<y/t> dimasukkan sebagai input

# DAFTAR PUSTAKA

[www.cplusplus.com](http://www.cplusplus.com)

Kristanto, Andi. 2003. Struktur Data dengan C++. Yogyakarta : Graha Ilmu.

Raharjo, B dan Heryanro, I. 2003. Pemrograman Borland C++ Builder. Bandung : Informatika.

## TENTANG PENULIS



Agus waluyo merupakan pria kelahiran kendal, 27 februari 1992 silam. perjalanan studinya dimulai dari bangku SD 04 Singorojo lulus tahun 2004 kemudian melanjutkan sekolahnya ke MTs NU 22 singorojo, setelah tamat pada tahun 2007 perjalanannya dilanjutkan ke SMA Negeri 1 Boja dan dinyatakan lulus pada tahun 2010. Saat ini Agus Waluyo masih tercatat sebagai mahasiswa semester 3 di universitas stikubank semarang (UNISBANK) jurusan Teknik Informatika.

kritik dan saran serta pertanyaan bisa melalui

[www.aguswaluyo27.blogspot.com](http://www.aguswaluyo27.blogspot.com)

[blogaguswaluyo@gmail.com](mailto:blogaguswaluyo@gmail.com)

[www.facebook.com/aguswaluy0](https://www.facebook.com/aguswaluy0)